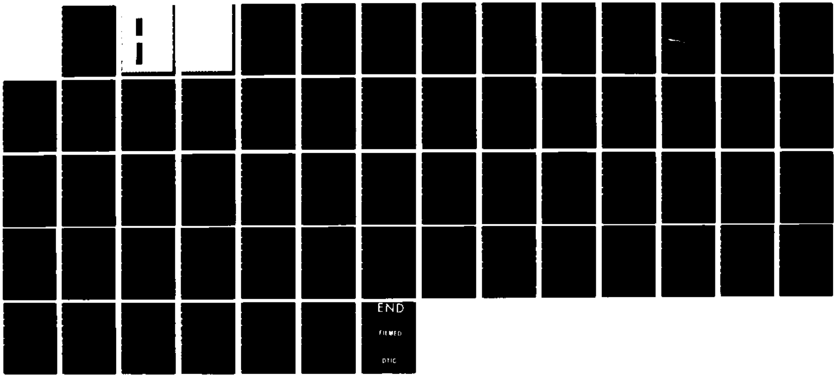
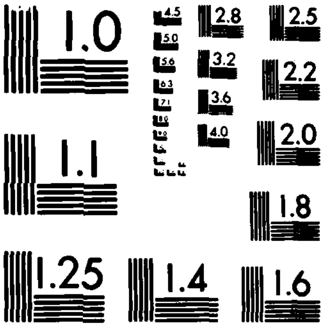
AD-A160 990 WESSEL: CODE FOR NUMERICAL SIMULATION OF 1/1
TWO-DIMENSIONAL TIME-DEPENDENT W... (U) MISSISSIPPI STATE
UNIV MISSISSIPPI STATE DEPT OF AEROPHYSICS A...
UNCLASSIFIED J F THOMPSON ET AL AUG 85 WES/TR/E-85-8 F/G 20/4 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

2

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Technical Report E-85-8	2. GOVT ACCESSION NO. AD-A160990	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (*and Subtitle*) WESSEL: CODE FOR NUMERICAL SIMULATION OF TWO-DIMENSIONAL TIME-DEPENDENT WIDTH-AVERAGED FLOWS WITH ARBITRARY BOUNDARIES		5. TYPE OF REPORT & PERIOD COVERED Final report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Joe F. Thompson, Robert S. Bernard		8. CONTRACT OR GRANT NUMBER(s) Contract No. DACW39-78-C-0054
9. PERFORMING ORGANIZATION NAME AND ADDRESS Mississippi State University, Department of Aerospace Engineering, Drawer A, Mississippi State, Mississippi 39762; US Army Engineer Waterways Experiment Station, Hydraulics Laboratory, PO Box 631, Vicksburg, Mississippi 39180-0631		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS CWIS No. 31604 (EWQOS Task IIIA.4)
11. CONTROLLING OFFICE NAME AND ADDRESS DEPARTMENT OF THE ARMY US Army Corps of Engineers Washington, DC 20314-1000		12. REPORT DATE August 1985
		13. NUMBER OF PAGES 55
14. MONITORING AGENCY NAME & ADDRESS(*if different from Controlling Office*) US Army Engineer Waterways Experiment Station Environmental Laboratory PO Box 631, Vicksburg, Mississippi 39180-0631		15. SECURITY CLASS (*of this report*) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (*of this Report*)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (*of the abstract entered in Block 20, if different from Report*)

18. SUPPLEMENTARY NOTES

Available from National Technical Information Service, 5285 Port Royal Road, Springfield, Virginia 22161.

19. KEY WORDS (*Continue on reverse side if necessary and identify by block number*)

Boundary-fitted coordinates	Reservoirs
Hydrodynamics	Selective withdrawal
Numerical modeling	Stratified flow

20. ABSTRACT (*Continue on reverse side if necessary and identify by block number*)

The code WESSEL solves the two momentum equations, the energy equation, and the continuity equation on a two-dimensional field with boundaries of arbitrary shape, including multiple inlets, outlets, and obstacles. The basis of this numerical solution is a boundary-fitted curvilinear coordinate system that allows all computation to be done on a rectangular field with a square grid in the transformed plane, regardless of the boundary shape and

(Continued)

20. ABSTRACT (Continued).

configuration in the physical plane. The finite difference solution is done in finite volume formulation. The solution is implicit in time, with the difference equations being solved simultaneously by SOR iteration at each time step. The code reads the boundary-fitted coordinate system from the output of the coordinate code WESCOR. The input allows any portions of the boundary (external or obstacles) to be designated as inlets, outlets, no-slip surfaces, or slip surfaces. Arbitrary specification of the variables on inlets and outlets is allowed. The output is in the form of field arrays and plots of the velocity components, pressure, and temperature. All computation is done in metric units, but the input and output units may be specified otherwise.

The WESSEL code permits analysis of hydrodynamics for a variety of applications to Civil Works projects.

PREFACE

The developments reported herein were sponsored by the Office, Chief of Engineers (OCE), US Army, as part of the Environmental and Water Quality Operational Studies (EWQOS) Program, under the work unit (CWIS 31604) entitled "Techniques to Meet Environmental Quality Objectives for Reservoir Releases." The effort was supported by Task IIIA.4, "Selective Withdrawal Characteristics of Various Outlet Configurations." The OCE Technical Monitors for EWQOS were Mr. Earl E. Eiker, Dr. John Bushman, and Mr. James L. Gottesman.

This project was conducted during the period June 1979 to September 1983 as part of a contract (DACW39-78-C-0054) with Mississippi State University (MSU) to develop a boundary-fitted-coordinate numerical hydrodynamic code for reservoir selective withdrawal. The research and development were carried out by Dr. Joe F. Thompson of the Department of Aerospace Engineering, MSU, and by Dr. Robert S. Bernard of the Reservoir Water Quality Branch (RWQB), Hydraulics Laboratory (HL), US Army Engineer Waterways Experiment Station (WES). Drs. Thompson and Bernard prepared this report.

Mr. H. B. Simmons, Chief, HL, and Mr. J. L. Grace, Jr., Chief of the Hydraulic Structures Division, directed the effort. Supervision was provided by Mr. J. P. Holland, Chief, RWQB, and Dr. D. R. Smith, former Chief of the RWQB. Mr. Mark S. Dortch, formerly of the RWQB, and Dr. Billy H. Johnson of the Mathematical Modeling Group, HL, monitored the contract. Program Manager of EWQOS was Dr. J. L. Mahloch, WES.

During the preparation of this report, COL Tilford C. Creel, CE, and COL Robert C. Lee, CE, were Commanders and Directors of WES and Mr. F. R. Brown was Technical Director. At the time of publication, COL Allen F. Grum, USA, was Director and Dr. Robert W. Whalin was Technical Director.

This report should be cited as follows:

Thompson, J. F., and Bernard, R. S. 1985. "WESSEL: Code for Numerical Simulation of Two-Dimensional Time-Dependent Width-Averaged Flows with Arbitrary Boundaries," Technical Report E-85-8, US Army Engineer Waterways Experiment Station, Vicksburg, Miss.

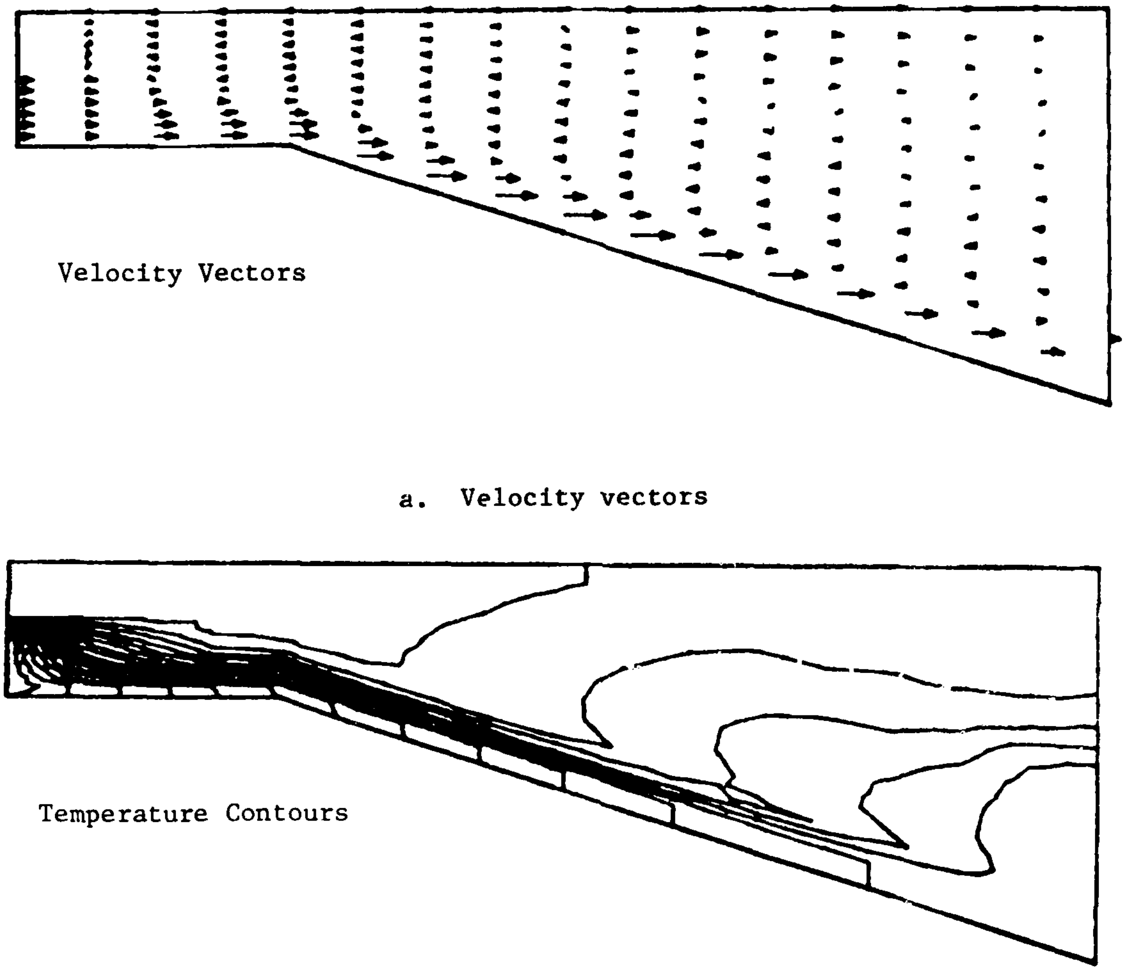
CONTENTS

WESSEL: CODE FOR NUMERICAL SIMULATION OF TWO-DIMENSIONAL TIME-DEPENDENT WIDTH-AVERAGED FLOWS WITH ARBITRARY BOUNDARIES

PART I: INTRODUCTION

1. The code WESSEL solves the two momentum equations, the energy equation, and the continuity equation on a two-dimensional field with boundaries of arbitrary shape, including multiple inlets, outlets, and obstacles. The basis of this numerical solution is a boundary-fitted curvilinear coordinate system that allows all computation to be done on a rectangular field with a square grid in the transformed plane, regardless of the boundary shape and configuration in the physical plane.

2. The finite difference solution is done in finite volume formulation, as discussed in detail in Thompson and Bernard (1985). The solution is implicit in time, with all the difference equations being solved simultaneously by SOR iteration at each time step. The code reads the boundary-fitted coordinate system from the output of the coordinate code WESCOR, described in Thompson (1983). The input allows any portions of the boundary (external or obstacles) to be designated as inlets, outlets, no-slip surfaces, or slip surfaces. Arbitrary specification of the variables on inlets and outlets is allowed. The output is in the form of field arrays and plots (Figure 1) of the velocity components, pressure, and temperature. All computation is done in metric units, but the input and output units may be specified otherwise. Appendixes A, B, and C contain input instructions for the hydrodynamic code (WESSEL), the contour plot code (CONTUR), and the vector plot code (VECTOR), respectively. Sample runstreams (input) are given in Appendix D.



a. Velocity vectors

b. Temperature contours

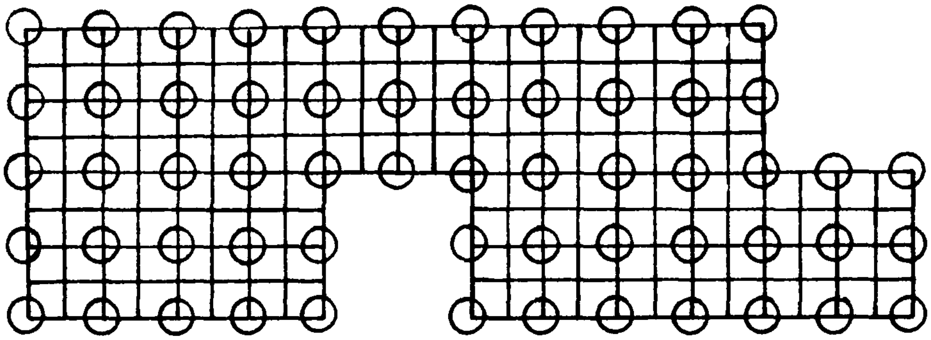
Figure 1. Sample output

PART II: CONFIGURATIONS

Computational Field

3. The computational field, i.e., the transformed region, is composed of contiguous rectangular blocks with a uniform square grid over the entire field, where the independent variables are the curvilinear coordinates (ξ,η)* designated in the code as the integers (I,J). As noted in Thompson and Bernard (1985), the increments, $\Delta\xi$ and $\Delta\eta$, are irrelevant since they cancel out of the difference equations, and hence are made unity for convenience. The extent of ξ and η on the computational field is from 1 to IMAX and JMAX, respectively. The dimensions of the field arrays are (0:IDIM,0:JDIM). All field arrays are extended one line beyond the computational field for convenience. Thus, the maximum values of IMAX and JMAX allowed are IDIM-1 and JDIM-1, respectively.

4. The coordinate system is generated with twice as many points in each direction as are used for the flow solution. Therefore, the coordinate arrays are dimensioned (0:ICIM,0:JCIM), where ICIM=2*IDIM-2 and JCIM=2*JDIM-2. The point in the coordinate arrays corresponding to the tield point (I,J) is thus (II,JJ), where II=2*I-1 and JJ=2*J-1. This correspondence is indicated in Figure 2.



0: solution points

Figure 2. Diagram of the general field

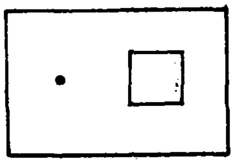
* For convenience, symbols are listed and defined in the Notation (Appendix E).

Point Classification

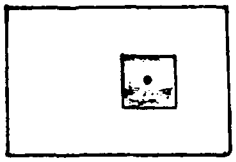
5. This code is applicable to a wide variety of configurations including multiple inlets, outlets, and obstacles, with all boundaries being of arbitrary shape. In order to achieve this versatility, it is necessary that the points be classified as to location on the various types of boundaries, in the free field or out of the computation region. This classification is done by associating a different name with each type of point as follows:

6. *Points not on boundaries.*

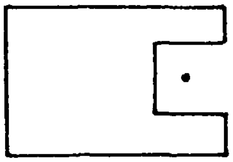
FIELD: field point (not on a boundary)



ØUT: point totally out of the computation region, i.e., inside an obstacle or beyond the outer boundary of the flow field

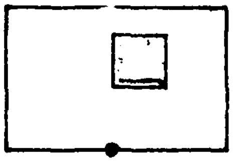


or

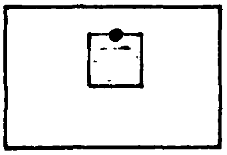


7. *Boundary points not on corners.*

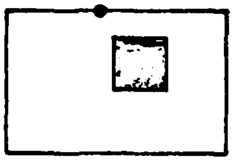
WALBØT: point on a lower field boundary with no-slip



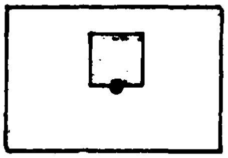
or



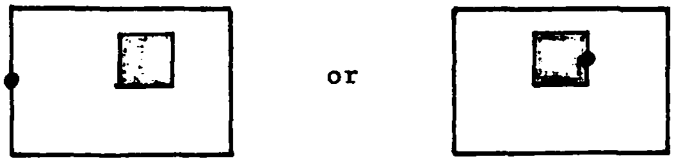
WALTØP: point on a top field boundary with no-slip



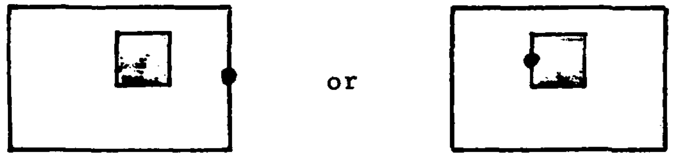
or



WALLEF: point on a left field boundary with no-slip



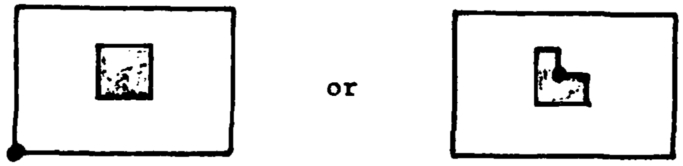
WALRIT: point on a right field boundary with no-slip



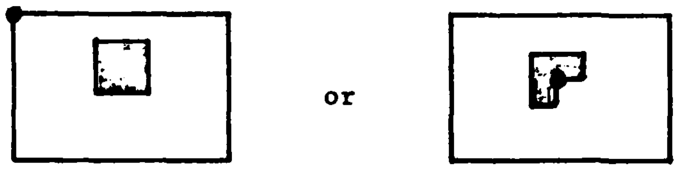
With the "WAL" in the above replaced by "SLI," the location is the same, but the boundary has slip. Similarly "IN" in place of "WAL" refers to inlet, "ØUT" to outlet, and "SUR" to free surface.

8. Corner points.

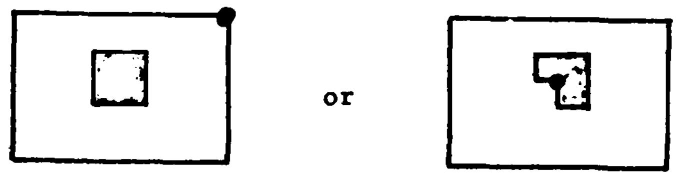
CAVWBL: point on bottom-left concave corner of field with no-slip



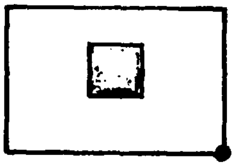
CAVWTL: point on top-left concave corner of field with no-slip



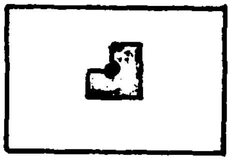
CAVWTR: point on top-right concave corner of field with no-slip



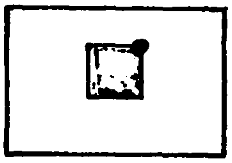
CAVWBR: point on bottom-right concave corner of field with no-slip



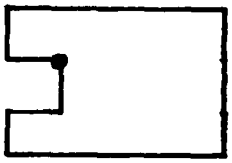
or



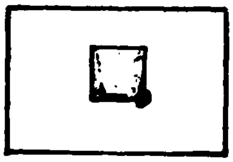
VEXWBL: point on bottom-left convex corner of field with no-slip



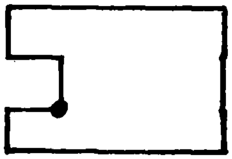
or



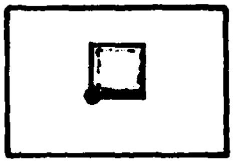
VEXWTL: point on top-left convex corner of field with no-slip



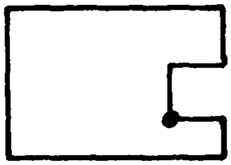
or



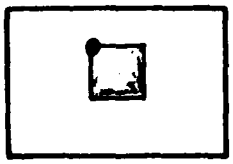
VEXWTR: point on top-right convex corner of field with no-slip



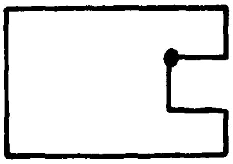
or



VEXWBR: point on bottom-right convex corner of field with no-slip



or



With the "W" in the preceding eight types replaced by "S," the reference is to a slip boundary.

9. <u>Point notation.</u> Each of the point type names is made an integer and is assigned a number. Thus ØUT=0, FIELD=25, etc. The types are also grouped as follows:

WALBØT=1		CAVWBL=5	
WALLEF=2		CAVWTL=6	
WALTØP=3		CAVWTR=7	
WALRIT=4	WALL=4	CAVWBR=8	WALCAV=8
VEXWBL=9		SLIBØT=13	
VEXWTL=10		SLILEF=14	
VEXWTR=11		SLITØP=15	
VEXWBR=12	WALVEX=12	SLIRIT=16	SLIP=16
CAVSBL=17		VEXSBL=21	
CAVSTL=18		VEXSTL=22	
CAVSTR=19		VEXSTR=23	
CAVSBR=20	SLICAV=20	VEXSBR=24	SLIVEX=24
INBØT=26		ØUTBØT=30	
INLEF=27		ØUTLEF=31	
INTØP=28		ØUTTØP=32	
INRIT=29	INLET=29	ØUTRIT=33	ØUTLET=33
SURBØT=34			
SURLEF=35			
SURTØP=36			
SURRIT=37	SURF=37		

Then if a point type is >ØUT and ≤WALL, the point must be on one of the no-slip walls. Similarly, if a point type is > FIELD and ≤INLET, the point must be on an inlet, etc.

PART III: NOTATION

10. FORTRAN variables and parameters are identified below, with the corresponding algebraic quantities in the equations of Thompson and Bernard (1985) shown in parentheses. The array dimensions are also given in parentheses for each type of array.

Main flow variable field arrays (0:IDIM,0:JDIM)*

VELX
: x and y velocity components (u,v)
VELY

PRES: difference between true pressure and hydrostatic pressure (P)

TEMP: temperature (T)

QWAL: boundary heat transfer (q_{wall})

Additional flow variable field arrays (0:IDIM,0:JDIM)

URØLD: $\frac{J}{\Delta t}\left[2(\rho u)^{n-1} - \frac{1}{2}(\rho u)^{n-2}\right]$ stored quantity from previous time steps for second-order time differencing

$\frac{J}{\Delta t}\left[(\rho u)^{n-1}\right]$ stored from previous time step for first-order time differencing

VRØLD: as URØLD, with u replaced by v

RØLD: as URØLD, with u replaced by 1 and J/Δt omitted

ERØLD: as URØLD, with u replaced by energy e

UACC: x-momentum equation acceleration parameter

VACC: y-momemtum equation acceleration parameter

TACC: energy equation acceleration parameter

TYPE: point type

STAT: hydrostatic pressure with constant density RHØREF (p_H)

WIDE: width (B)

* This format makes the array dimensions IDIM+1 , JDIM+1 .

ENGY: energy (e)

DIVEL: divergence of velocity ($\underset{\sim}{\nabla} \cdot \underset{\sim}{u}$)

CMASØ: mass residual

CMAS: as URØLD, with u replaced by mass residual, and J replaced by $J^2B(JD_t)$

Coordinate arrays (0:ICIM,0:JCIM)

XCØR
: cartesian coordinates (x,y)
YCØR

SIDE: point type array LSLIT of the coordinate code WESCØR

Eddy viscosity

CART: artificial viscosity $\left(\frac{|D|}{B}\right)$

RICH: Richardson number for turbulent eddy viscosity (R_i)

TURHØR: selector for horizontal turbulent eddy viscosity

TURBUL: turbulence indicator

CØFHØR: specified uniform horizontal turbulent eddy viscosity (D_{11})

TURVER: selector for vertical turbulent eddy viscosity

CØFVER: specified uniform vertical turbulent eddy viscosity (D_{22})

Run control parameters

STØRIT: first time step to be stored; later, next step to be stored

START: indicates type of start

STØINT: interval between stored time steps

ITERS: maximum number of iterations allowed

PARCØN: accepts partial convergence

STEPS: number of time steps to be run

STEP: time step number on file

Iteration tolerances

UTØL
VTØL
PTØL
TTØL

$\left.\begin{array}{l}\text{UTØL}\\ \text{VTØL}\\ \text{PTØL}\\ \text{TTØL}\end{array}\right\}$ iteration error tolerances for u , v , P , T

UNØRM: iteration error norm for x-momentum equation $(||\Delta u||)$

VNØRM: iteration error norm for y-momentum equation $(||\Delta v||)$

PNØRM: iteration error norm for pressure equation $(||\Delta P||)$

TNØRM: iteration error norm for energy equation $(||\Delta T||)$

(IUNØRM,JUNØRM): location of maximum x-momentum iteration error (substitute VNØRM, etc., for other norms)

Indices

D,ØD: diagonal and off-diagonal elements of stress and heat transfer terms

$\left.\begin{array}{l}\text{IMAX}\\ \text{JMAX}\end{array}\right\}$ field extent

(I,J): indices of calculation point

(II,JJ): indices for coordinate arrays corresponding to (I,J)

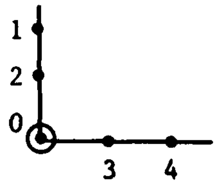
Solution options

ENEREQ: allows energy equation to be deleted

TIMFAC(TIMØRD,L): coefficients in time-derivative for order TIMØRD:

$$\frac{\Delta f}{\Delta t} = \frac{\text{TIMFAC(TIMØRD,1)}f^{n} + \text{TIMFAC(TIMØRD,2)}f^{n-1} + \text{TIMFAC(TIMØRD,3)}f^{n-2}}{\Delta t}$$

CAVFAC(CAVØRD,L): coefficients in extrapolation at concave corners for order CAVØRD



$$f_0 = \text{CAVFAC(CAVØRD,1)}*(f_2 + f_3) + \text{CAVFAC(CAVØRD,2)}*(f_1 + f_4)$$

CØNVEC: selector for convective differences

TIMØRD: time derivative order

CAVØRD: concave corner extrapolation order

FLØBAL: activator for flow balance

FLØW: activator for inflow or outflow

STRAT: selector for initial stratification

ISTRAT: ξ-line defining initial stratification

RHØS: table of density defining initial stratification

TEMS: temperature table defining initial stratification

ELEV: elevation table defining initial stratification

Acceleration parameters

UACTØL
VACTØL
TACTØL
iteration error tolerances for variable acceleration parameters for u , v , T , respectively

CØSI $\cos \frac{\pi}{IMAX+1}$
CØSJ $\cos \frac{\pi}{JMAX+1}$
used in calculation of variable acceleration parameters

BACC: boundary temperature acceleration parameter

PACC: pressure acceleration parameter

VELACC: switch for variable acceleration parameter for momentum equations

TEMACC: switch for variable acceleration parameter for energy equation

VELXAC
VELYAC
TEMPAC
u , v , T acceleration parameters, respectively, when variable parameters are not calculated

PRESAC: pressure acceleration parameter (input value)

BØUNAC: boundary temperature acceleration parameter

Solution quantities

RHØREF: reference density used for calculation of hydrostatic pressure (ρ_0)

GX
GY
gravity vector (g_1,g_2)

FLØIN: total inlet flow rate

FLØØUT: total outlet flow rate

DELTAT: time step (Δt)

VELREF
TEMREF
PREREF
reference values of (u,v) , T , p , respectively

GRAX
GRAY
gravity unit vector

UGEN
VGEN
PGEN
TGEN
WGEN
general values for u , v , P , T , B , respectively, over entire field

(IREF,JREF):
(XREF,YREF):
location of PREREF, reference pressure for calculation of hydrostatic pressure

GRAC: acceleration of gravity

Output options

VUNITS
PUNITS
TUNITS
output units for (u,v) , P , and T , respectively

CUNITS: output units for time

EXTØUT: switch for output of field extrema

MAPØUT: switch for output of field maps

MAP(5): field maps desired

RITØUT: switch for printed output of field at selected steps

RITERR: switch for printed output of iteration error norms

RITINT: switch for printed output of initial field solution

LABEL: output label

ACCPRT: switch for printed output concerning variable acceleration parameters

RITEXT: switch for printed output of outlet variables

Computation site arrays (25)

U	u		
V	v		
P	P		
T	T		
RHØ	ρ		
RHØU	ρu		
RHØV	ρv		
RHØE	ρe		
ENERGY	e		
MU	μ		
KC	κ		
Q	heat source flux		
POINT	point type		
SII	σ_{11}, etc.	(25,2)	second index refers to diagonal (1) or off-diagonal (2)
SIIBAR	$\tilde{\sigma}_{11}$, etc.	(25,2)	
Q1	q_1, etc.	(25,2)	
Q1BAR	$\tilde{q}_1$, etc.	(25,2)	
DSXI	(2)	stress terms in momentum equation, conduction terms in energy equation	
DSET	(2)		
UXXI	ux_ξ	etc. for v ..., T ..., P ...	
UYXI	uy_ξ		
UXET	ux_η		
UYET	uy_η		
UBAR	$\tilde{u}$		
VBAR	$\tilde{v}$		
X	x		
Y	y		
XXI	x_ξ		
YXI	y_ξ		

XET	x_η
YET	y_η
JCB	$J = x_\xi y_\eta - x_\eta y_\xi$
ALPHA	$\alpha = x_\eta{}^2 + y_\eta{}^2$
GAMMA	$\gamma = x_\xi{}^2 + y_\xi{}^2$
INFCEN,JNFCEN	index increments for neighboring solution points
INSCEN,JNSCEN	index increments for neighboring coordinate points
Q	heat flux (q)

Other variables

D11,D22	horizontal and vertical eddy viscosity (D_{11},D_{22})
C11,C22	horizontal and vertical eddy conductivity (C_1,C_2)
MASDEF	mass residual (D)
FLØRAT	ratio of inlet to outlet flow rate before balance correction
PTØT	full pressure (p)
AJAC	Jacobian (J)
UACMIN,UACMAX, UACAVG	minimum, maximum, and average acceleration parameter for velocity (etc. for V and T)
VANØRM	iteration error norm for x-velocity acceleration parameter (etc. for V and T)
ØVLEX ØVELY ØTEMP ØENGY ØCMAS	previous time step values of u , v , T , e , D , respectively
NPT	total number of field points
TIME	time (t)
TIM	time in output units (t)
MU11,MU22	viscosity including turbulence ($\mu + \rho D_{11}$,$\mu + \rho D_{22}$)
MU1122	average of MU11 and MU22

KC11,KC22	conductivity including turbulence $(\kappa + \rho \frac{\kappa}{\mu} C_1, \kappa + \rho \frac{\kappa}{\mu} C\ )$
DPXI,DPET	pressure terms in momentum equations
UCØLD,VCØLD	velocity values at previous iterate
XXID	x_ξ
XETD	x_η
RHØB	ρB
DEPTH	depth to computation site below top (d)
HEIGHT	height of computation site above bottom (h)
FULDEP	top-to-bottom distance (d_0)
VELC	velocity magnitude at computation site ($\lvert\underset{\sim}{u}\rvert$) (etc. for N , S , E , W)
VELHØR,VELVER	horizontal and vertical derivatives of velocity magnitude ($\lvert\underset{\sim}{u}\rvert_x$, $\lvert\underset{\sim}{u}\rvert_y$)
HORCEL,VERCEL	horizontal and vertical cell width ($\Delta x, \Delta y$)
RHØVER	vertical derivative of density (ρ_y)
DMXI,DMET	ξ and η convective flux terms (ρBfu and ρBfv) (Suffix C indicates coefficient of value of f at
DMXIC,DMETC	computation site. Entire term is DMXI + DMXIC*f_c, etc.)
EMASDEF	mass residual correction term (zero for nonconservative convective terms, equal to MASDEF for conservative terms)
CENFAC	total coefficient of central value factored from all terms
UTEMP	intermediate x-velocity value (etc. for v , P , T)
XXID	x_ξ (etc. for Y)
XETD	x_η
ALFA	α
GAMA	γ
BETA	$\beta = x_\xi x_\eta + y_\xi y_\eta$
DELT	$x_\xi y_\eta + x_\eta y_\xi$

XXIXI	$x_{\xi\xi}$	(etc. for Y)
XETET	$x_{\eta\eta}$	
XXIET	$x_{\xi\eta}$	
DDX	$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta}$	
DDXX	$\overline{DDXX}$	
DDXY	$\overline{DDXY}$	
DDYX	$\overline{DDYX}$	
DDYY	$\overline{DDYY}$	
DDXYX	$\overline{DDXYX}$	
DDXYY	$\overline{DDXYY}$	
UXI	u_ξ	(etc. for v , B , P)
UET	u_η	
UX	Ju_x	
UY	Ju_y	
BRXI	$(B\rho)_\xi$	
BRET	$(B\rho)_\eta$	
BRXIXI	$(B\rho)_{\xi\xi}$	
BRETET	$(B\rho)_{\eta\eta}$	
BRXIET	$(B\rho)_{\xi\eta}$	
BRX	$J(B\rho)_x$	
BRY	$J(B\rho)_y$	
BRXX	$(B\rho)_{xx}$	
BRYY	$(B\rho)_{yy}$	
BRXY	$(B\rho)_{xy}$	
BRTXI	$(B\rho_t)_\xi$	
BRTET	$(B\rho_t)_\eta$	
BRTX	$J(B\rho_t)_x$	

BRTY	$J(B\rho_t)_y$	
DXI	$(\underset{\sim}{\nabla}\cdot\underset{\sim}{u})_\xi$	
DET	$(\underset{\sim}{\nabla}\cdot\underset{\sim}{u})_\eta$	
DXIXI	$(\underset{\sim}{\nabla}\cdot\underset{\sim}{u})_{\xi\xi}$	
DETET	$(\underset{\sim}{\nabla}\cdot\underset{\sim}{u})_{\eta\eta}$	
DXIET	$(\underset{\sim}{\nabla}\cdot\underset{\sim}{u})_{\xi\eta}$	
DX	$J(\underset{\sim}{\nabla}\cdot\underset{\sim}{u})_x$	
DY	$J(\underset{\sim}{\nabla}\cdot\underset{\sim}{u})_y$	
DLAP	$J^2\nabla^2(\underset{\sim}{\nabla}\cdot\underset{\sim}{u})$	
UXIXI	$u_{\xi\xi}$	(etc. for V)
UETET	$u_{\eta\eta}$	
UXIET	$u_{\xi\eta}$	
ULAP	$J^2\nabla^2 u$	
DVEL	$J\underset{\sim}{\nabla}\cdot\underset{\sim}{u}$	
RUXI	$\left(\frac{uD}{J}\right)_\xi$	
RUET	$\left(\frac{uD}{J}\right)_\eta$	
RUX	$J\left(\frac{uD}{J}\right)_x$	
RVY	$J\left(\frac{vD}{J}\right)_y$	
RHS	$BJ^2\nabla^2 p$	
ARTVC	artificial viscosity $\left(\text{CART*}\ \frac{\lvert D\rvert}{B}\right)$	
ARTCC	artificial conductivity $\left(\text{CART*}\ \frac{\kappa}{\mu}\ \frac{\lvert D\rvert}{B}\right)$	

DISP viscous dissipation term $\left(J\sigma_{ij}\ \frac{\partial(Bu_i)}{\partial x_j}\right)$

DILT volume dilation term $Jp\left(\frac{\partial(Bu_j)}{\partial x_j}\right)$

PART IV: OPERATION

11. The code starts by reading the namelist BEGIN, which contains a flag START which indicates an initial start, a continuation of a partially converged iteration, or a continuation of time steps.

Initial Start

12. For an original start, the namelist PARAM is read next, and subroutine SETVAL is called to read the remainder of the input. This subroutine first reads the label, the extent of the coordinate arrays (IMAXC,JMAXC), and the coordinate point type array (SIDE, the LSLIT of WESCØR) from the output of the coordinate system code WESCØR. The extent of the field arrays (IMAX,JMAX) is then calculated, and the field point type array TYPE is initialized to ØUT. For free field points, i.e., points in the flow field and not on a boundary, TYPE is then set to FIELD, and for points on any boundary, TYPE is set to the default, WALL, indicating a no-slip boundary. The coordinate system is then read from the output of WESCØR into the coordinate arrays XCØR and YCØR.

13. The boundary types are designated next by input as follows. For each boundary segment (outer or obstacle) that is not to be no-slip, namelist INPUT is read specifying the boundary type (INLET, OUTLET, SURFACE, or SLIP) and the indices, (I1,J1) and (I2,J2), of the segment ends. All points on that segment are then set to this type in TYPE. Points on all boundary segments not designated otherwise by input are typed no-slip by default, as noted above, except that it is also possible to change this default to slip by the input. After all of the boundary designations have been read, the code determines if the segments are bottom, left, right, or top relative to the free field, and locates concave and convex corners, placing the appropriate type classification in TYPE for each boundary point.

14. The field arrays for velocity, pressure, temperature, energy, width, and wall heat transfer (VELX, VELY, PRES, TEMP, ENGY, WIDE, and

QUAL) are initialized to the default values. The general field values and the reference pressure and position then may be changed from the default values by a read of the namelist INPUT for each quantity to be changed. Units are read with the values, and the code converts all quantities to metric. The units of the coordinate system must be identified, or metres will be used, and the time step is set. Boundary values on any boundary segment are then specified by a read of the namelist INPUT identifying the segment, the quantity to be specified, the values, and the units. The general field values remain on any boundary segment that is not changed in this manner. Initial density (or temperature) stratification may be specified either on a designated ξ-line or with an input table of density (or temperature) versus elevation.

15. Specification of temperature on a boundary segment causes that segment to operate with a fixed temperature. Specification of heat transfer on a segment causes it to operate with that heat transfer. No specification causes a segment to operate as an adiabatic wall.

16. Subroutine SETVAL then completes its setup of the field and boundaries by calculating the components of the gravity vector, the reference values and their locations, the hydrostatic pressure, and the energy for all points, and by initializing the acceleration parameter arrays to the default values. The reference velocity is the square root of the sum of the squares of the maximum magnitudes of the components on the field and boundary. The reference temperature and density are the general values on the field. The reference pressure and the reference position are items previously input with the general values. Hydrostatic pressure is calculated with the initial density distribution by integrating from the reference position, where the full pressure is equal to the reference pressure, with constant density equal to the reference density.

17. The main program then initializes the rest of the field arrays, setting the velocity divergence, DIVEL, and the mass residual terms, CMAS and CMASØ , to zero, and setting the previous time flow variables to the initial values. The initial solution is then printed if desired, completing the initialization of an original start.

Continued Iteration Start

18. If the start is to be made from a partially converged solution, the read of namelist BEGIN is followed by a read of the partially converged solution from file 13. A read of the namelist PARAM, so that parameters can be changed if desired, then completes the setup for continuation of the iteration.

Continued Time Steps Start

19. Finally, if the start is to begin from the solution at some previous time, the read of namelist BEGIN is followed by a read of the solution at this time from file 13. Namelist PARAM is then read, completing the setup for continuation of the time steps.

Setup for Time Step

20. After any of these three possible starting modes, the convergence tolerances for velocity, pressure, and temperature are calculated from the reference values. These tolerances are calculated by multiplying the input tolerances by the corresponding reference values, except that the pressure tolerance is based on the maximum of the reference pressure and the dynamic pressure ρV^2 , calculated from the reference density and velocity. The reference values and tolerances are then printed, and the code is then ready to proceed with the time steps. If the step is the first time step from an initial start, the code elects first-order time difference expressions regardless of the input specification of the order. The input specification is used at all subsequent time steps.

21. If the start is not from a partially converged solution, before proceeding with the calculation for the solution at $t + \Delta t$, the current values of velocity, temperature, energy, and the mass residual term CMASØ at t are written in the corresponding field arrays,

ØVELX, ØVELY, ØTEMP, ØENGY, and ØCMAS. The inhomogeneous terms (URØLD, VRØLD, ERØLD, RØLD) and the mass residual term CMAS are calculated for each FIELD point. If nonconservative convective differences are used, the density is omitted from URØLD, VRØLD, ERØLD, and RØLD. This completes the setup for the calculation of the solution at the new time step.

Iteration

Boundaries update

22. The iteration then proceeds. At each iteration, the norms are zeroed, all of the boundary values are updated, and then all of the field values are updated. In each case the updated values replace the current values in the arrays as they are calculated, in accordance with Gauss-Seidel iteration. Values on no-slip or slip boundaries are updated by calling subroutine WALLS from the appropriate entry point, BØTTØM, TØP, LEFT, or RIGHT, relative to the free field. (Note that a point on the top of an obstacle calls BØTTØM, etc.) Concave and convex corners, whether no-slip or slip, are treated by calling the subroutines CAVCØR and VEXCØR, respectively. Inlet, outlet, and surface points are updated by calling subroutine INFLØ, ØUTFLØ, or SRFACE, respectively. The energy (ENGY) is calculated from the temperature at all of the boundary points. Consequently, this routine is called through the appropriate entry point, after each call to INFLØ, ØUTFLØ, or SRFACE. After the complete boundary update, the normal velocity components on all outlets or inlets may be adjusted so that the total outflow exactly equals the total inflow by calling subroutine BALØUT or BALIN.

Field update

23. The updating of the field values is done by a call to DIFFEQ at each FIELD point. After the field has been updated, the error norms are checked for convergence to the specified tolerances. If convergence has been attained, then the previous time step values are retrieved, and the solution is written on the restart file 11. The solution is then

printed and stored on the solution file 12 if the current time step is designated for output. The solution then proceeds to the next time step unless the prescribed number of steps has been executed, in which case the solution stops.

Completion

24. If the maximum number of iterations allowed is reached without convergence, there are two possible alternatives. If the input has so specified, the code will interpret this as convergence and proceed with the output and continue to the next time step as above. Otherwise, the partially converged solution is written on the restart file 11. The values at the previous time step are retrieved and written also. In this case, the solution stops.

PART V: SUBROUTINES

BLØCK Data

25. This data set contains the default values.

Subroutine AITKEN

26. This subroutine produces a new iteration for a Newton-Raphson iterative solution of a nonlinear equation.

Subroutine CAVCØR

27. This subroutine calculates the temperature on a concave corner by linear or quadratic extrapolation along the two sides forming the corner.

Subroutine DIFFEQ

28. This subroutine calculates the values of the pressure, temperature, and velocity components at each FIELD point. The routine first places the appropriate values of the coordinates of the points surrounding the point of calculation in the arrays X and Y. The coordinate derivatives (XXI, etc.) are then calculated where needed and, from these, the Jacobian and α and γ are calculated. Next, the width, velocity, temperature, energy, and pressure are placed where needed. The density is calculated from the temperature.

29. The viscosity, the turbulent eddy viscosity, and the artificial viscosity (if used) are then calculated where needed. The viscosity is calculated directly from the temperature. The turbulent eddy viscosity may be an input constant or may be calculated from either of two models. The artificial viscosity is calculated from the mass residual. The quantities u and v, the contravariant velocity components, are the calculated and the mass residual is evaluated.

30. The routine then calls subroutines CHØRIN, THERMØ, and MØMENT to calculate the pressure, temperature, and velocity, respectively.

Pressure

31. Subroutine CHØRIN calculates updated pressure via the Chorin scheme (Thompson and Bernard 1985). The routine first evaluates the mass residual using first-order one-sided differences. The new pressure is set equal to the old pressure, minus the mass residual multiplied by the pressure acceleration parameter and the geometric coefficient derived in Thompson and Bernard (1985). The change from the previous iteration is noted, and the new pressure is placed in the pressure field array PRES. This routine is called before *and* after calling MØMENT, in order to improve convergence.

Temperature

32. Subroutine THERMØ updates the temperature from the energy equation. (There is an input provision for skipping this routine if the density is to be held constant.) This routine first calculates the conductivity from the temperature where needed and then evaluates the turbulent eddy thermal diffusivity and the artificial conductivity (if used) from the eddy and artificial viscosities calculated above. The heat conduction terms are then evaluated, with terms containing the temperature at the update point being identified as diagonal terms and the remaining terms as off-diagonal terms by the indices D and ØD , respectively. Finally, the energy flux terms are evaluated, using a choice of central, upwind, or ZIP differences (Thompson and Bernard 1985), isolating any terms containing the energy at the update point. Second-order central differences are used for all derivatives except for upwind convective terms, which are first-order.

33. The new temperature at the update point is then calculated through a false position iteration, since the time derivative and possibly the energy flux terms contain the energy at the update point, while the conduction terms contain the temperature at this point. This Newton-Raphson iteration is done by using subroutine AITKEN. If variable acceleration parameters are being used, this parameter is updated

by calling subroutine TEMAC after the iteration for the temperature at the update point has converged. The temperature is then updated, the change from the previous iteration is noted, the new temperature is placed in the temperature field array, TEMP, and the new energy is calculated from the new temperature.

Velocity

34. Subroutine MØMENT calculates the velocity components from the momentum equations. The routine first evaluates the stress terms, identifying the terms containing the velocity at the update point as diagonal terms (index D) and the remainder as off-diagonal terms (index ØD). The momentum flux terms are then evaluated using a choice of central, upwind, or ZIP differences, again identifying terms containing the velocity at the update point. The pressure terms are then evaluated. Second-order central differences are used for all deriviatives except for pressure gradients and upwind convective terms. The new velocity is calculated, with all terms involving the component being evaluated at the update point factored together. The mass residual correction term uD is omitted if nonconservative convective differences are used. If variable acceleration parameters are to be used, subroutine VELAC is then called to update this parameter. The new velocity is then calculated. The change from the previous iteration is noted, and the new velocity is placed in the velocity field arrays, VELX and VELY.

Subroutine INFLØ

35. Thi, subroutine calculates the temperature of an inlet by setting the value equal to the adjacent value inside the field. If FLØW is input as 'ØUTFLØW,' the inlet velocity is extrapolated in the same way. The routine also calculates the mass flow rate into the inlet by summing the product of the appropriate contravariant velocity component and the density.

Subroutine MAPS

36. This subroutine prints maps of various field quantities for inspection. These may be elected by the input parameters in the array MAP:

a. Point type map--shows the type for each point, i.e., field, inlet, outlet, etc.

b. Velocity map--shows interval contours of the velocity magnitude by assigning a different number from 0 to 9 to each point, depending on the ratio of the velocity from the minimum value on the field to the difference between the maximum and minimum values on the field. Intervals of 0.1 from 0.0 to 1.0 are indicated by the numbers 0-9.

c. Temperature map--shows interval contours of the temperature in the same manner as described above for the velocity.

d. Pressure map--shows interval contours of the difference between the pressure and the hydrostatic pressure, as for the velocity.

e. Density map--shows interval contours of the density, as for the velocity.

Subroutine ØUTFLØ

37. This subroutine calculates the temperature of an outlet by setting the value equal to the adjacent value inside the field. If FLØW is input as 'INFLØW,' the outlet velocity is extrapolated in the same way. The routine also calculates the mass flow rate out of the outlet in the manner described above for INFLØ.

Subroutine SETVAL

38. This subroutine reads the input and sets up the initial field as described above.

Subroutine SRFACE

39. This subroutine calculates the velocity components and the temperature on a simulated free surface (without movement of the surface) by setting these values equal to the values at the adjacent points inside the field.

Subroutine TEMAC

40. This subroutine calculates the locally optimum acceleration parameter for the energy equation. This parameter depends on the local convective and diffusive terms, and underrelaxation is used unless diffusion dominates in both directions. The change from the values at the previous iteration is noted, and the new value is placed in the field array TACC.

Subroutine VELAC

41. This subroutine calculates the locally optimum acceleration parameters for the momentum equations. This parameter depends on the local convective and diffusive terms, and underrelaxation is used unless diffusion dominates in both directions. The change from the values at the previous iteration is noted, and the new values are placed in the field arrays, UACC and VACC.

Subroutine VEXCØR

42. This subroutine calculates the temperature and velocity components at a convex corner. This is done by calling subroutine WALLS, with the appropriate entry point, for each of the two sides forming the corner, and then averaging the results.

Subroutine WALLS

43. This subroutine calculates the temperature and pressure on no-slip and slip boundaries. It also evaluates the slip velocity on slip boundaries. There are four entry points for boundaries on the BØTTØM, TØP, LEFT, or RIGHT relative to the free field. Note that the top of an obstacle and the bottom of the entire region are classified as BØTTØM.

44. The routine first places the coordinates of neighboring points in the arrays X and Y and calculates the coordinate derivatives. The values of all quantities needed at the surrounding points are placed in the appropriate arrays, and the derivatives are evaluated using second-order central differences in the field and along the boundary, except at corners where first-order one-sided differences are used. Such one-sided differences are also used for derivatives coming off the wall.

45. After the heat conduction terms are evaluated, either the new temp:rature is calculated from the specified wall heat transfer or the wall heat transfer is calculated from the specified wall temperature. The wall pressure is evaluated by subroutine CHØRIN. If the boundary is a slip boundary, then the new velocity components are calculated with the vorticity and normal velocity set to zero.

46. The new temperature and velocity components are calculated using a specified uniform acceleration parameter, the change from the previous iteration is noted, and the new values are placed in the field arrays.

Subroutine XTREMA

47. This subroutine calculates and prints the extrema of the velocity, temperature, and pressure.

Subroutine BALØUT

48. This subroutine adjusts the velocities on all the outlets to balance the inlet and outlet flow rates. This is done by multiplying the velocity normal to the outlets by the ratio of the inlet flow rate to the outlet flow rate. New velocities are then calculated (for outlets only) using the adjusted normal velocity and the unchanged tangential velocity. Use of this routine is an input option.

Subroutine BALIN

49. This subroutine is the same as BALØUT, but is applied to inlets instead of outlets. Use of this routine is an input option.

PART VI: PLOT CODES

50. These codes, called VECTØR and CØNTUR, respectively, plot velocity vectors at each point or selected points in the field, and the contours of density, temperature, and pressure.

Contour Plots (CØNTUR)

51. The contour plots are done as described below. The code was written by Thompson, but the description is taken from Thames (1975).

Determination of contours in the ξ,η plane

52. Let $\phi = \phi(\xi,\eta)$ be a function defined in the region $\overline{D}*$ possessing continuous second derivatives. Since $\overline{D}*$ is closed and bounded, let $m(\phi) \equiv \min\{\phi(\xi,\eta) \mid [\xi,\eta]\varepsilon\ \overline{D}*\}$ and $M(\phi) \equiv \max\{\phi(\xi,\eta) \mid \xi,\eta]\varepsilon\ D*\}$. If Φ is a number such that $m(\phi) \leq \Phi \leq M(\phi)$, then we define the Φ-contour of $\phi(\xi,\eta)$, $C_T(\Phi)$, as the set

$$C_T(\Phi) = \{[\xi,\eta] \mid [\xi,\eta]\varepsilon\ \overline{D}* \text{ and } \phi(\xi,\eta) = \Phi\}$$

Graphically, $C_T(\Phi)$ is the curve created by the intersection of the graph of $\phi(\xi,\eta)$ and the plane $\phi(\xi,\eta) = \Phi$. For plotting convenience, the curve is usually projected onto the (ξ,η) plane. These ideas are illustrated in Figure 3.

53. Now suppose that $\phi(\xi,\eta)$ is known only in a discrete fashion. That is, let the net function $\phi_{i,j} \equiv \phi(\xi_i,\eta_j)$ be known on the discrete set $\overline{D}** = \{[\xi_i,\eta_j] \mid \xi_i = i-1$ for $1 \leq i \leq IMAX$ and $\eta_j = j-1$ for $1 \leq j \leq JMAX\}$. (The fact that $\phi_{i,j}$ may only be an approximation to $\phi(\xi,\eta)$ is immaterial to the current discussion.) If similar definitions for $m(\phi)$ and $M(\phi)$ are made for $\phi_{i,j}$ on the set $D**$, then the Φ-contour of $\phi_{i,j}$, denoted $C_T(\Phi)$ again, can be defined as

$$C_T(\Phi) = \{[\bar{\xi}_k, \bar{\eta}_k] \mid 0 \leq \bar{\xi}_k \leq \text{IMAX-1};\ 0 \leq \bar{\eta}_k \leq \text{JMAX-1};$$

$$\bar{\phi}(\bar{\xi}_k, \bar{\eta}_k) \approx \Phi;\ k=1,2,\ldots,N;\ N \geq 2\}$$

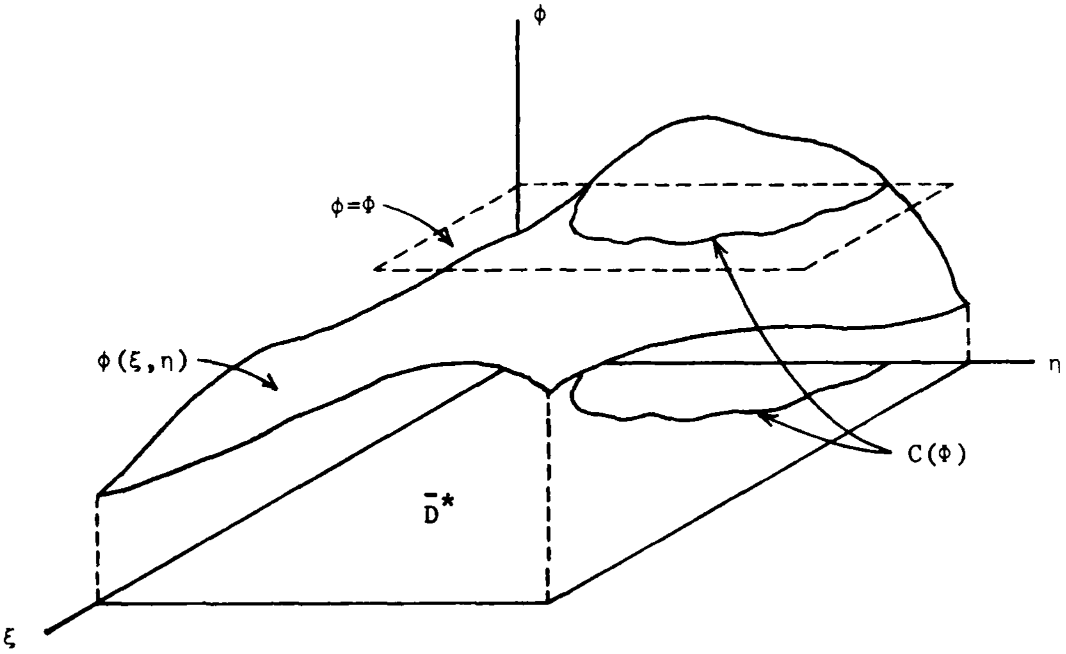


Figure 3. Contour - Φ of $\phi(\xi,\eta)$

The bars over ξ_k, η_k, and ϕ are to indicate that $[\bar{\xi}_k, \bar{\eta}_k]$ may not be an element of $\bar{D}^{**}$ and that $\bar{\phi}$ is not necessarily one of the values of the net function $\phi_{i,j}$. The discrete representation is shown in Figure 4.

54. Numerically, the import of the above discussion is that interpolation between the points of the discrete set $\bar{D}^{**}$ is required to determine $C_T(\Phi)$. Consider a portion of grid $\bar{D}^{**}$ as shown in Figure 5a where $1 \leq I1 < I2 \leq IMAX$ and $1 \leq J1 < J2 \leq JMAX$. Each

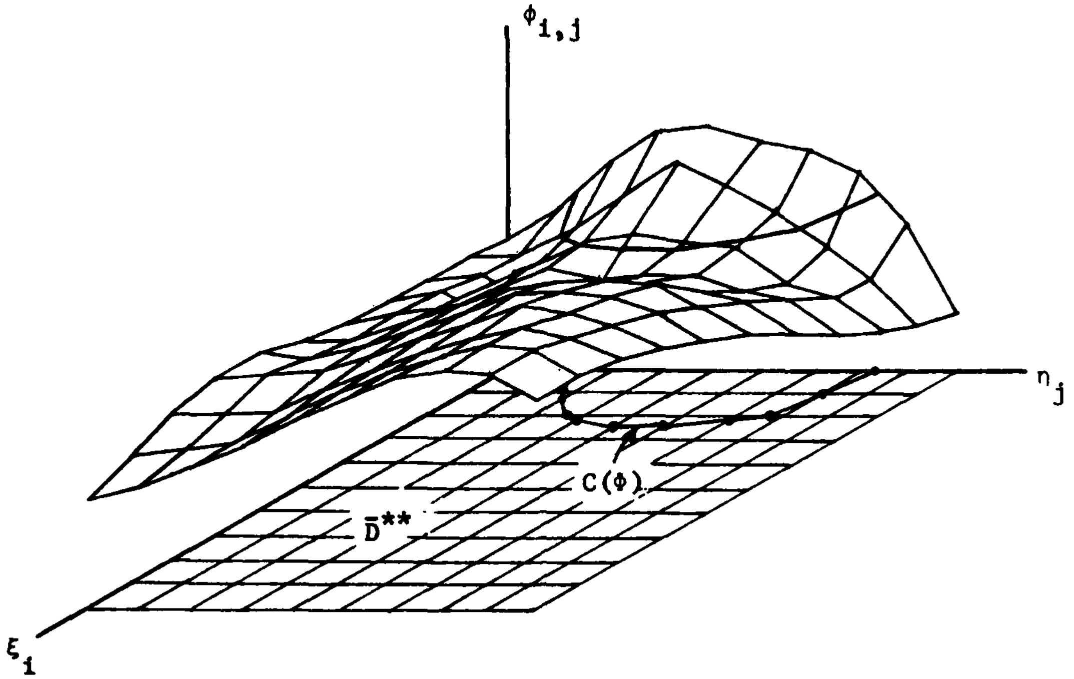
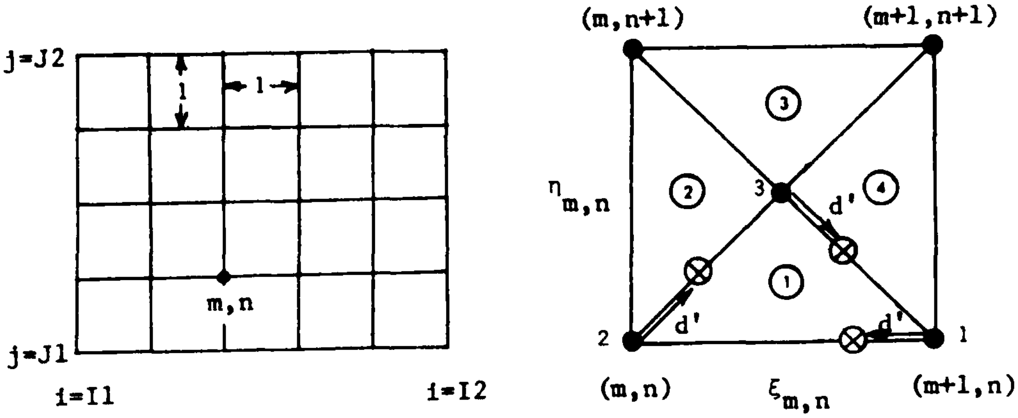


Figure 4. Contour of $\phi(\xi_i,\eta_j)$



a. Portion of grid $\overline{D}$**

b. Block (m,n)

Figure 5. Sample grid in set $\overline{D}$**

grid block is labeled by the (i,j) coordinates of the lower left-hand corner of the block. Block (m,n) is shown on a larger scale in Figure 5b.

55. In order to improve the plotted resolution, consider subdividing each block into four triangles, as shown. The value of ϕ at $(m + \frac{1}{2}, n + \frac{1}{2})$ is taken as the four-point average

$$\phi_{m + \frac{1}{2},n + \frac{1}{2}} = (\phi_{m,n} + \phi_{m+1,n} + \phi_{m,n+1} + \phi_{m+1,n+1})/4$$

A local (ξ,η) coordinate is affixed to each grid block as demonstrated in Figure 5b. In order to standardize the interpolation procedures, a local (ζ,μ) coordinate system is also placed on each of the subtriangles as illustrated in the series of drawings given in Figure 6.

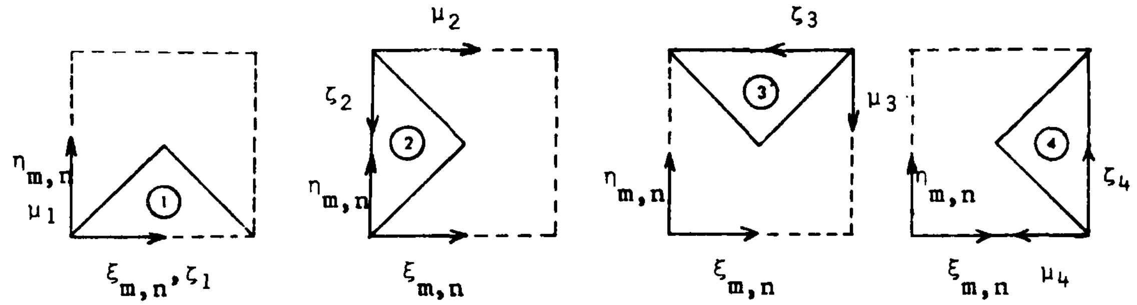


Figure 6. Local coordinate systems for triangles

56. Interpolation is carried out on each of the four triangles for each grid block in the segment ($I1 \leq i \leq I2$, $J1 \leq j \leq J2$) of the set $\overline{D}$** specified. In particular, interpolation is performed along each of the three sides of each of the triangles if the contour value Φ lies between the values at the ends of the sides. Let d' be the directed distance from a given triangle vertex to the point on the triangle side where the contour intersects that side (denoted by $\otimes$). This distance is illustrated in Figure 5b for triangle ① and is defined in an analagous manner for the other triangles. If Φ_1 is the value of $\phi_{i,j}$ at a particular triangle vertex and Φ_2 is the value of $\phi_{i,j}$ at the other end of the side, then d' may be expressed as

$$d' = (\text{side length})\ (\Phi - \Phi_1)/(\Phi_2 - \Phi_1)$$

For example, along side 1-2 of triangle ①, d' is given by

$$d'_{1-2} = (1.0)(\phi - \phi_{m+1,n})/(\phi_{m,n} - \phi_{m+1,n})$$

57. Noting that the sides of the triangle have lengths 1.0, $1.0/\sqrt{2}$, and $1.0/\sqrt{2}$, the contour intersection can be expressed in the local triangle coordinates as

Side	ζ_ℓ	μ_ℓ
1-2	1-d	0
2-3	d/2	d/2
3-1	(1+d)/2	(1-d)/2

where $d \equiv (\phi - \Phi_1)/(\Phi_2 - \Phi_1)$ and where $\ell = 1, 2, 3,$ or 4 denotes the triangle number.

58. Once the contour intersections have been determined in the local triangle coordinates (ζ_ℓ, μ_ℓ), they must be transformed to the grid block coordinates $(\xi_{m,n}, \eta_{m,n})$. This is done in the conventional fashion using orthogonal rotation matrices. If $[\zeta_{\ell,p}, \mu_{\ell,p}]$ are the coordinates of an intersection in triangle ℓ, then

$$\begin{bmatrix} (\xi_{m,n})_{\ell,p} \\ \\ (\eta_{m,n})_{\ell,p} \end{bmatrix} = \underline{A}_\ell \begin{bmatrix} \zeta_{\ell,p} \\ \\ \mu_{\ell,p} \end{bmatrix}$$

where

$$\underline{A_1} \equiv \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \underline{A_2} \equiv \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

$$\underline{A_3} \equiv \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \quad \underline{A_4} \equiv \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Note that up to three contour intersections can occur for each triangle (i.e., one on each side). Finally, the point $[(\xi_{m,n})_{\ell,p}, (\eta_{m,n})_{\ell,p}]$ is transformed to (ξ,η) coordinates by a simple linear transformation producing an element$[\overline{\xi}_k, \overline{\eta}_k]$ of the set $C_T(\Phi)$

Transformation to the physical plane

59. Since contours in the (ξ,η) plane are of little interest, $C_T(\Phi)$ must be transformed to the physical plane. This is made possible through the use of the coordinate transformation functions $x(\xi_i,\eta_j)$ and $y(\xi_i,\eta_j)$. Again, interpolation is required since almost all elements of $C_T(\Phi)$ are not elements of $\overline{D}^{**}$ on which the discrete functions $x(\xi_i,\eta_j)$ and $y(\xi_i,\eta_j)$ are defined. As illustrated in Figure 7, this implies a double linear interpolation must be performed. If $[\overline{\xi}_k, \overline{\eta}_k]$ denotes an element of $C_T(\Phi)$, the first step is to locate the ξ and η values bracketing $\overline{\xi}_k$ and $\overline{\eta}_k$. Denoting these by ξ_i, ξ_{i+1} and η_j, η_{j+1} as shown in the figure, the values of $\overline{x}_k$ and $\overline{y}_k$ are calculated as follows:

$$\overline{x}_k = (\overline{\eta}_k - \eta_j)(X_{j+1} - X_j)/(\eta_{j+1} - \eta_j) + X_j$$

where

$$X_j = (\bar{\xi}_k - \xi_i)(x_{i+1,j} - x_{i,j})/(\xi_{i+1} - \xi_i) + x_{i,j}$$

$$X_{j+1} = (\bar{\xi}_k - \xi_i)(x_{i+1,j+1} - x_{i,j+1})/(\xi_{i+1} - \xi_i) + x_{i,j+1}$$

for all k=1,2,...,N . Similar expressions are used to calculate $\bar{y}_k$.

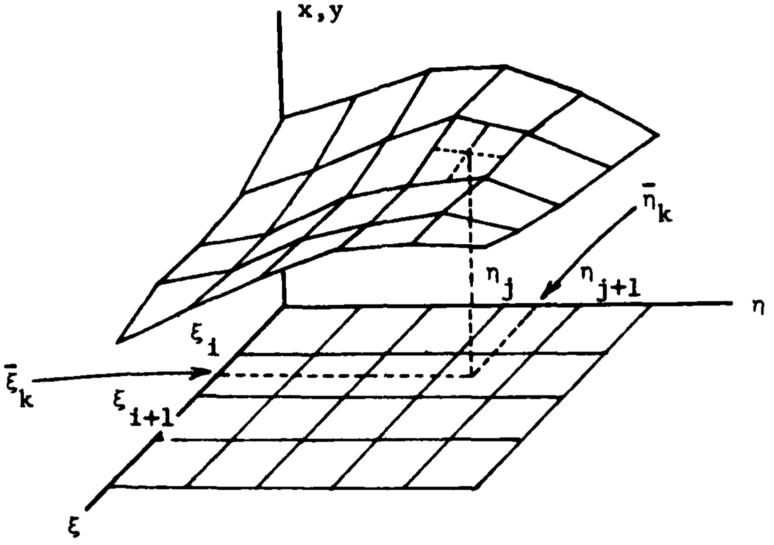


Figure 7. Interpolation for x and y

Vectors (VECTØR)

60. The velocity vectors are formed by simply drawing a line from each point, the length of which is proportional to the velocity magnitude and the orientation is that of the velocity vector. An arrowhead is added to the tip of the line.

REFERENCES

Thames, F. C. 1975. "Numerical Solution of the Incompressible Navier-Stokes Equations About Arbitrary Two-Dimensional Bodies," Ph. D. Dissertation, Mississippi State University, Mississippi State, Miss.

Thompson, J. F. 1983. "A Boundary-Fitted Coordinate Code for General Two-Dimensional Regions with Obstacles and Boundary Intrusions," Technical Report E-83-8, US Army Engineer Waterways Experiment Station, Vicksburg, Miss.

Thompson, J. F., and Bernard, R. S. 1985. "Numerical Modeling of Two-Dimensional Width-Averaged Flows Using Boundary-Fitted Coordinate Systems," Technical Report E-85-9, US Army Engineer Waterways Experiment Station, Vicksburg, Miss.

APPENDIX A: WESSEL INPUT INSTRUCTIONS

WESSEL Input Instructions

```
************************************************************************
*
*********************** W E S S E L ************************************
*
************************************************************************
*
** INPUT INSTRUCTIONS **************************************************
*
************************************************************************
*
* FILES % FOR CONTINUATION OF TIME STEPS, OR CONTINUATION OF ITERATION
*         ON A PARTIALLY CONVERGED TIME STEP, THE INITIAL SOLUTION IS
*         READ FROM UNIT 13 UNFORMATTED.
*
*         THE LAST TIME STEP, OR A PARTIALLY CONVERGED TIME STEP, IS
*         WRITTEN ON UNIT 11 UNFORMATTED FOR RESTART.
*
*         THE TIME STEPS SELECTED FOR STORAGE ARE WRITTEN UNFORMATTED
*         ON UNIT 12.
*
*         COORDINATE SYSTEM IS READ UNFORMATTED FROM UNIT 10.
*         (AS WRITTEN BY CODE 'WESCOR' )
*
************************************************************************
*
* INPUT %  (QUANTITIES NOT INDICATED OTHERWISE ARE FLOATING POINT.)
*
*-----------------------------------------------------------------------
*
*** NAMELIST $BEGIN %
*
*       START - START SELECTOR %
*               'INITIAL' % INITIAL START.
*               'NEXT'    % START FROM CONVERGED TIME STEP.
*               'RESTART' % CONTINUE ITERATION ON PARTIALLY CONVERGED TIME STEP.
*
*-----------------------------------------------------------------------
*
*** NAMELIST $PARAM %
*
*       NAMELIST ITEMS % (ANY ORDER, OMISSIONS ALLOWED)
*                        (DEFAULT VALUES IN PARATHENSES AT END)
*
*       FLOW   - SELECTOR FOR FLOW SPECIFICATION. ('INOUT' )
*                'INFLOW'  % INFLOW SPECIFIED.
*                'OUTFLOW' % OUTFLOW SPECIFIED.
*                'INOUT'     INFLOW AND OUTFLOW SPECIFIED.
*
*       ENEREQ - SELECTOR FOR ENERGY EQUATION.  ('YES' OR 'NO').  ('YES')
*                ('NO' OMITS ENERGY EQUATION, KEEPING UNIFORM TEMPERATURE)
*
*       PARCON - ACCEPTOR FOR PARTIAL CONVERGENCE.  ('YES' OR 'NO').  ('NO')
*                ('YES' ACCEPTS PARTIAL CONVERGENCE AT MAXIMUN NUMBER OF )
*                ( ITERATIONS AND PROCEEDS TO NEXT TIME STEP.            )
*
*       STEPS  - NUMBER OF TIME STEPS.          (INTEGER). (1)
*
*       STORIT - FIRST STEP TO BE STORED.       (INTEGER). (1)
*       STOINT - INTERVAL BETWEEN STORED STEPS. (INTEGER). (1)
*                (1 STORES EVERY STEP, 2 EVERY SECOND STEP, ETC.)
*
*       STRAT  - SELECTOR FOR INITIAL STRATIFICATION.  ('NONE')
*                'NONE'  % NO STRATIFICATION.
*                'LINE'  % STRATIFICATION ACCORDING TO DENSITY OR
*                          TEMPERATURE INPUT ON XI-LINE ISTRAT.
*                'TABLE' % STRATIFICATION ACCORDING TO INPUT TABLE OF
*                          DENSITY OR TEMPERATURE VS ELEVATION.
*
*       ISTRAT - XI-LINE DEFINING INITIAL STRATIFICATION WHEN
*                STRAT='LINE'.  (1)
*                 (IRRELEVANT IF STRAT='NONE' OR 'TABLE')
*
*       UTOL   - CONVERGENCE TOLERANCE FOR X-VELOCITY.  (1.E-4)
*       VTOL   -                          Y-VELOCITY.  (1.E-4)
*       PTOL   -                          PRESSURE.    (1.E-4)
*       TTOL   -                          TEMPERATURE. (1.E-4)
*                (FRACTIONS OF MAXIMUM VALUES)
*
*       UACTOL - CONVERGENCE TOLERANCE FOR X-MOMENTUM ACC.PARA. (1.E-4)
*       VACTOL -                          Y-MOMENTUM ACC.PARA. (1.E-4)
*       TACTOL -                          ENERGY ACC.PARA.     (1.E-4)
*
*       VELXAC - ACCELERATION PARAMETER FOR X-MOMENTUM.           (1.)
*       VELYAC -                           Y-MOMENTUM.           (1.)
*       PRESAC -                           CHORIN PRESSURE EQ.   (0.8)
*       TEMPAC -                           ENERGY.               (1.)
*       BOUNAC -                           BOUNDARY TEMPERATURE. (1.)
*
*       VELACC - VARIABLE ACCELERATION PARAMETER ACTIVATOR FOR MOMENTUM. (.TRUE.)
*       TEMACC -                                              ENERGY.   (.TRUE.)
*                (LOGICAL).  (.TRUE. ACTIVATES VARIABLE PARAMETERS.)
*
*       DELTAT - TIME STEP.  (1.)
*
*       CUNITS - OUTPUT UNITS FOR TIME.        ('SEC')
*       VUNITS -                  VELOCITY.    ('MPS')
*       PUNITS -                  PRESSURE.    ('PASCALS')
*       TUNITS -                  TEMPERATURE. ('K')
*                (SEE LIST OF UNIT CHOICES BELOW. DEFAULT IS METRIC UNITS)
*
*       TIMORD - ORDER OF TIME DERIVATIVE.      (1)
*       CAVORD -          CORNER EXTRAPOLATION. (1)
*                (INTEGER, 1 OR 2 )
*
*       ITERS  - MAXIMUM NUMBER OF ITERATIONS. (INTEGER). (1)
*
*       GRAX   - GRAVITY UNIT VECTOR X-COMPONENT. ( 0.)
*       GRAY   -                     Y-COMPONENT. (-1.)
*
*       EXTOUT - ACTIVATOR FOR OUTPUT OF FIELD EXTREMA. ('NO')
*       MAPOUT -                              MAPS.    ('NO')
*                ('YES' OR ' NO')
*
*       RITINT - ACTIVATOR FOR PRINT OF INITIAL SOLUTION.   ('NO')
*       RITOUT -                       TIME STEP SOLUTION. ('NO')
*       RITEXT -                       SOLUTION ON ALL OUTLETS. ('NO')
*
*       RITERR -                       ITERATION NORMS.    ('YES')
*                ('YES' OR 'NO')
*
*       MAP(I) - ACTIVATORS FOR FIELD MAPS.  ('NONE')
*                'TYPE'        % POINT TYPE MAP.
*                'VELOCITY'    % VELOCITY MAP.
*                'PRES'    % PRESSURE MAP.
*                'TEMP' % TEMPERATURE MAP.
*                'DENS'     % DENSITY MAP.
*                  ('NONE' GIVES NO MAPS.)
*
*       CONVEC - SELECTOR FOR CONVECTIVE DERIVATIVES.  ('UPWIND')
*                CONSERVATIVE     % 'UPWIND' OR ' CENTRAL' .
*                NON-CONSERVATIVE % 'NONUP' OR 'NONCEN' .
*                ZIP              % 'ZIP' .
*
*       IREF,JREF - INDICES OF ELEVATION REFERENCE.  (1,1)
*
*       CART   - COEFFICIENT IN ARTIFICIAL VISCOSITY.  (0.)
*                (0.0 FOR NONE.  NOMINAL NON-ZERO VALUE IS 1.0 )
*
*       TURHOR - SELECTOR FOR HORIZONTAL TURBULENCE.  ('NONE')
*                'NONE'     % NO TURBULENCE.
*                'CONSTANT' % UNIFORM EDDY VISCOSITY = COFHOR.
*                'EDINGER'  % EDINGER MODEL.
*                'KENT'     % KENT MODEL.
*
*       TURVER - SELECTOR FOR VERTICAL TURBULENCE%  ('NONE')
*                 'NONE'     % NO TURBULENCE.
*                 'CONSTANT' % UNIFORM EDDY VISCOSITY = COFVER.
*                 'EDINGER'  % EDINGER MODEL.
*                 'KENT'     % KENT MODEL.
*
*       COFHOR - COEFFICIENT FOR HORIZONTAL TURBULENCE. (1.)
*                 (IRRELEVANT IF TURHOR NOT EQUAL 'CONSTANT')
*
*       COFVER - COEFFICIENT FOR VERTICAL TURBULENCE. (1.)
*                 (IRRELEVANT IF TURVER NOT EQUAL 'CONSTANT')
*
*       FLOBAL - SELECTOR FOR FLOWRATE BALANCE. ('YES' OR 'NO').  ('NO')
*                 WHEN FLOW='INFLOW' THIS CAUSES ALL OUTLET NORMAL VELOCITIES TO
*                 BE MULTIPLIED BY RATIO OF TOTAL INLET FLOWRATE TO TOTAL OUTLET
*                 FLOWRATE.
*                 WHEN FLOW='OUTFLOW' THIS CAUSES ALL INLET NORMAL VELOCITIES TO
*                 BE MULTIPLIED RATIO OF TOTAL OUTLET FLOWRATE TO TOTAL INLET
*                 FLOWRATE.
*                 WHEN FLOW='INOUT' THIS OPTION IS IRRELEVANT.
*
*       ACCPRT - SELECTOR FOR ACCELERATION PARAMETER PRINT. ('YES' OR 'NO').
*
*---------------------------------------------------------------------
*
*** BOUNDARY TYPE NAMELIST $INPUT % (ANY ORDER, OMISSIONS ALLOWED)
*
*               BEGIN WITH A NAMELIST $INPUT WITH ITEM='BOUND' .
*
*               THEN USE A SEPARATE NAMELIST $INPUT FOR EACH BOUNDARY SEGMENT.
C
*               CLOSE WITH A NAMELIST $INPUT WITH ITEM='END' .
*
*       ITEM - BOUNDARY TYPE %   (DEFAULT IS NO-SLIP)
```

```
*
*              'INLET'    % INFLOW BOUNDARY. (VELOCITY & TEMPERATURE
*                                              SPECIFIED OR EXTRAPOLATED)
*
*              'OUTLET'   % OUTFLOW BOUNDARY. (VELOCITY & TEMPERATURE
*                                              SPECIFIED OR EXTRAPOLATED)
*
*              'SURFACE.'  % FREE SURFACE. (EXTRAPOLATED VELOCITY AND TEMPERATURE)
*
*              'SLIP'     % SLIP BOUNDARY. (ZERO VORTICITY & NORMAL VELOCITY)
*
*              'ALL SLIP'% CHANGES BOUNDARY DEFAULT FROM NO-SLIP TO SLIP.
*
*       (I1,J1) & (I2,J2) - INDICES OF ENDS OF BOUNDARY SEGMENT. (INTEGER)
*
*--------------------------------------------------------------------
*
*** GENERAL VALUE NAMELIST $INPUT % (ANY ORDER, OMISSIONS ALLOWED)
*
*                 BEGIN WITH A NAMELIST $INPUT WITH ITEM='GENERAL'
*
*                 THEN USE A SEPARATE NAMELIST $INPUT FOR EACH QUANTITY.
*
*                 CLOSE WITH A NAMELIST $INPUT WITH ITEM='END'
*
*       ITEM =
*
*              'VELX'     - X-VELOCITY  (1.)
*
*              'VELY'     - Y-VELOCITY  (1.)
*
*              'PRES'     - DIFFERENCE FROM REFERENCE PRESSURE  (0.)
*
*              'REFPRES'  - REFERENCE PRESSURE (AT IREF,JREF)   (1 ATM)
*
*              'TEMP'     - TEMPERATURE  (273.15)
*
*              'DENS'     - DENSITY  (1000.0)
*
*              'HEAT'     - HEAT TRANSFER  (0.)
*
*              'WIDTH'    - WIDTH  (1.)
*
*              'COORD'    - SERVES TO CHANGE COORDINATE UNITS.  (METRIC)
*                              (VALUE IS IRRELEVANT.)
*
*       VALUE - GENERAL VALUE OF QUANTITY.
*
*       UNITS =
*
*               'FPS'   (FEET/SEC)           ...VELOCITY
*               'MPH'   (MILES/HOUR)
*               'KNOTS' (NAUTICAL MILES/HOUR)
*
*               'ATM'   (ATMOSPHERES)  ........PRESSURE
*               'PSI'   (LB/SQ.IN.)
*               'PSF'   (LB/SQ.FT.)
*
*               'C'    (CELSIUS)  .............TEMPERATURE
*               'F'    (FAHRENHEIT)
*               'R'    (RANKINE)
*
*               'PCF'   (LB/CU.FT.)        ......DENSITY
*               'SCF'   (SLUGS/CU.FT.)
*               'GML'   (GRAMS/MILLILITER)
*               'GCC'   (GRAMS/CU.CM.)
*
*               'BTUSIS' (BTU/SQ.IN./SEC)  ....HEAT TRANSFER
*               'BTUSFS' (BTU/SQ.FT./SEC)
*
*               'FEET'         ...............WIDTH
*               'MILES'
*               'YARDS'
*               'NAUTICAL MILES'
*
*               'FEET'     ...................COORDINATES
*               'MILES'
*               'YARDS'
*               'NAUTICAL MILES'
*
*---------------------------------------------------------------------
*
*** SECTION VALUE NAMELIST $INPUT % (ANY ORDER, OMISSIONS ALLOWED)
*
*                      BEGIN WITH A NAMELIST $INPUT WITH ITEM='SECTION'
C
*                      THEN USE A SEPARATE NAMELIST $INPUT FOR EACH QUANTITY.
C
*                      CLOSE WITH A NAMELIST $INPUT WITH ITEM='END'
*
*       ITEM =
*
*              'VELX'  - X-VELOCITY  (1.)
*
*              'VELY'  - Y-VELOCITY  (1.)
*
*              'PRES'  - DIFFERENCE FROM REFERENCE PRESSURE  (1 ATM)
*
*              'TEMP'  - TEMPERATURE  (273.15)
*
*              'DENS'  - DENSITY  (1000.)
*
*              'HEAT'  - HEAT TRANSFER  (0.)
*
*              'WIDTH' - WIDTH  (1.)
*
*              'DENTAB' - INITIAL STRATIFICATION TABLE % DENSITY.
*              'TEMTAB' -                                 TEMPERATURE.
*              'ELETAB' -                                 ELEVATION.
*                (IRRELEVANT UNLESS 'TRAT='TABLE'. )
*                (USE 'DENTAB' OR 'TEMTAB', NOT BOTH.)
*
*       VALUES - VALUES OF QUANTITY ON SECTION.
C
*       (I1,J1) & (I2,J2) - INDICES OF CORNERS OF SECTION. (INTEGER)
*
*       JEL - NUMBER OF ENTRIES IN INITIAL STRATIFICATION TABLE.
*              (IRRELEVANT UNLESS STRAT='TABLE')
*
*       UNITS - UNITS OF QUANTITY.  (SEE LIST ABOVE)
```

APPENDIX B: CONTUR INPUT INSTRUCTIONS

CONTUR Input Instructions

```
100=C*****************************************************************
110=C*
120=C********************** C O N T U R *************************************
130=C*
140=C*****************************************************************
150=C
160=C  CONTOUR PLOT CODE   -  MISSISSIPPI STATE UNIVERSITY , 1982
170=C
180=C                         U.S. ARMY ENGINEER WATERWAYS EXPERIMENT STATION
190=C                         VICKSBURG, MISSISSIPPI
200=C
210=C*****************************************************************
220=C
230=C INPUT :  (QUANTITIES NOT INDICATED OTHERWISE ARE FLOATING POINT.)
240=C
250=C** NAMELIST $PARAM :
260=C
270=C        NAMELIST ITEMS : (ANY ORDER, OMISSIONS ALLOWED)
280=C                         (DEFAULT VALUES IN PARENTHESES AT END)
290=C
300=C        NXI    - NUMBER OF XI INDICES TO BE SCANNED. (INTEGER)
310=C                 (DEFAULT IS ALL)
320=C
330=C        NETA   - NUMBER OF ETA INDICES TO BE SCANNED. (INTEGER)
340=C                 (DEFAULT IS ALL)
350=C
360=C        ISYM   - SYMBOL AND LINE CONTROL. ZERO IF NO SYMBOLS ON THE
370=C                 CONTOUR LINE ARE DESIRED. A VALUE OF 3 GIVES SYMBOLS
380=C                 ON EVERY INTERSECTION, 2 OR 1 GIVES FEWER SYMBOLS. A
390=C                 NEGATIVE VALUE SUPRESSES THE CONNECTING LINE. SYMBOL
400=C                 USED IS SYMBOL#(NCON-1).  (INTEGER)
410=C                 (DEFAULT IS NO SYMBOLS)
420=C
430=C        NSTEPS - NUMBER OF TIME STEPS. (INTEGER)  (1)
440=C
450=C        SIZE   - PLOT SIZE IN Y-DIRECTION IN INCHES.  (8.)
460=C
470=C        SIZRAT - RATIO OF PLOT LENGTH IN X-DIRECTION TO SIZE. (1.)
480=C
490=C        XMIN,XMAX - CONTOUR PLOT LIMITS IN X DIRECTION.
500=C                   (DEFAULT IS ALL)
510=C
520=C        YMIN,YMAX - CONTOUR PLOT LIMITS IN Y DIRECTION.
530=C                   (DEFAULT IS ALL)
540=C
550=C        IGRID  - PUTS GRID LINES OF 1.(IGRID+1) WEIGHT AT EACH TIC
560=C                 MARK LOCATION. NEGATIVE INTEGER PRODUCES HALF-WEIGHT
570=C                 GRID LINES AT EACH TIC MARK LOCATION AND -IGRID NUM-
580=C                 BER OF FOURTH-WEIGHT LINES EVENLY SPACED BETWEEN)
590=C                 (INTEGER)   (DEFAULT IS NO GRID)
600=C
610=C        DAX,   - HORIZONTAL AND VERTICAL AXES TIC MARK LENGTHS IN INCHES.
620=C        DOX      (DEFAULT IS NO TIC MARKS)
630=C
640=C        OA,OB  - HORIZONTAL AND VERTICAL AXES MINIMUM VALUES.
650=C                 (DEFAULT IS NO AXES NUMBERS )
660=C
670=C        OAI,   - HORIZONTAL AND VERTICAL AXES NUMBER INTERVAL.
680=C        OBI      (DEFAULT IS NO AXES NUMBERS )
690=C
700=C** NAMELIST $INPUT :
710=C
720=C        NAMELIST ITEMS : (ANY ORDER, OMISSIONS ALLOWED)
730=C                         (A SEPARATE NAMELIST $INPUT FOR EACH VARIABLE TO BE CONTOURED)
740=C
750=C            ITEM =
760=C
770=C                  'DENSITY'  - DENSITY
780=C
790=C                  'ENERGY'   - ENERGY
800=C
810=C                  'PRESSURE' - DIFFERENCE FROM HYDROSTATIC PRESSURE
820=C
830=C                  'TEMPERAT' - TEMPERATURE
840=C
850=C            NUM - NUMBER OF CONTOURS FOR THIS VARIABLE. (INTEGER)  (1)
```

```
C
C             UNITS =                (DEFAULT IS METRIC)
C
C                      'PCF'  - POUNDS/CUBIC FOOT     ..DENSITY
C                      'SCF'  - SLUGS/CUBIC FOOT
C                      'GCC'  - GRAMS/CUBIC CENTIMETER
C
C                      'FT-LBF'  - FOOT POUNDS ........ENERGY
C                      'FT-LBL'  - FOOT POUNDALS
C                      'BTU'     - BTU
C                      'CAL'     - CALORIES
C                      'ERG'     - ERGS
C                      'KCAL'    - KILOCALORIES
C                      'KWH'     - KILOWATT HOURS
C                      'WH'      - WATT HOURS
C
C                      'ATM'  - ATMOSPHERES       ......PRESSURE
C                      'PSI'  - POUNDS/SQUARE INCH
C                      'PSF'  - POUNDS/SQUARE FOOT
C
C                      'C'  - CELSIUS  ................TEMPERATURE
C                      'F'  - FAHRENHEIT
C                      'R'  - RANKINE
C
C             VALUES(NUM) - CONTOUR VALUES.
C
C** IF LESS THAN FOUR VARIABLES ARE CONTOURED, FOLLOW THE CONTOUR
C** NAMELISTS WITH AN $INPUT NAMELIST CONTAINING ONLY ITEM='END'.
C
C********************************************************************
C
C  TIME STEPS ARE READ UNFORMATTED FROM UNIT 11, AS WRITTEN BY CODE 'WESSEL'.
C
C  COORDINATE SYSTEM IS READ UNFORMATTED FROM UNIT 10, AS WRITTEN BY CODE 'WESCOR'.
C
C****************************************************************
```

APPENDIX C: VECTOR INPUT INSTRUCTIONS

VECTOR Input Instructions

```
C*******************************************************************
C
C********************** V E C T O R ********************************
C
C*******************************************************************
C   VECTOR PLOT CODE  -  MISSISSIPPI STATE UNIVERSITY , 1982
C
C                       U.S. ARMY ENGINEER WATERWAYS EXPERIMENT STATION
C                       VICKSBURG, MISSISSIPPI
C
C*******************************************************************
C
C  SOLUTION IS READ UNFORMATTED FROM UNIT 11, AS WRITTEN BY CODE 'WESSEL'
C
C  COORDINATE SYSTEM IS READ UNFORMATTED FROM UNIT 10, AS WRITTEN BY CODE 'WESCOR'
C
C****************************************************************
C INPUT :  (QUANTITIES NOT INDICATED OTHERWISE ARE FLOATING POINT.)
C
C** NAMELIST $PARAM :
C
C       NAMELIST ITEMS : (ANY ORDER, OMISSIONS ALLOWED)
C                       (DEFAULT VALUES IN PARENTHESES AT END)
C
C       NSTEPS - TOTAL NUMBER OF STEPS TO BE PLOTTED. (INTEGER) (1)
C
C       NXI    - NUMBER OF XI INDICES TO BE SCANNED. (INTEGER)
C                (DEFAULT IS ALL)
C
C       NETA   - NUMBER OF ETA INDICES TO BE SCANNED. (INTEGER)
C                (DEFAULT IS ALL)
C
C       NSKXI  - SKIP INTERVAL FOR XI.   (INTEGER)  (1)
C                (1 SCANS EVERY LINE, 2 EVERY SECOND LINE, ETC.)
C
C       NSKETA - SKIP INTERVAL FOR ETA.  (INTEGER)  (1)
C                (SEE NSKXI)
C
C       SIZE   - PLOT SIZE IN Y-DIRECTION IN INCHES. (8.)
C
C       SIZRAT - RATIO OF PLOT LENGTH IN X-DIRECTION TO SIZE. (1.)
C
C       AB     - ARROWHEAD ANGLE IN DEGREES. (20.)
C
C       AD     - ARROWHEAD LENGTH IN INCHES. (0.0875)
C
C       S2     - VELOCITY SCALE FACTOR , FRACTION OF SIZE PER UNIT VELOCITY. (1.)
C
C       XMIN,XMAX -  PLOT LIMITS IN X DIRECTION.
C                    (DEFAULT IS ALL)
C
C       YMIN,YMAX -  PLOT LIMITS IN Y DIRECTION.
C                    (DEFAULT IS ALL)
C
C*******************************************************************
```

APPENDIX D: SAMPLE RUNSTREAMS

WESSEL

```
00100 /JOB
00110 MJLQXXX(T500,CM500,P02,STCA1)
00120 USER(XXXXXX,XXXXXX)
00130 ***   XWESSEL
00150 FETCH(DN=$BLD,GDN=ORSBWSL,UN=XXXXXX,DS=CI)
00160 FETCH(DN=FT10,GDN=RSBCDO,DS=CI)
00170 LDR(MAP=FULL,E=1)
00180 EXIT(U)
00190 COST(LO=F)
00200 EXIT(U)
00230 STORE(DN=FT11,GDN=RSBDT6,UN=XXXXXX,DS=CI)
00240 EXIT(U)
00250 STORE(DN=FT12,GDN=RSBSOL6,UN=XXXXXX,DS=CI)
00260 EXIT(U)
00270 LOGFILE(L=WESLDAY)
00280 STORE(DN=WESLDAY,UN=XXXXXX,DS=FF,DT=C)
00281 /EOR
00290 E$BEGIN START='INITIAL' $
00300 E$PARAM STEPS=500    , TIMORD=2     , PARCON='YES' , ITERS=50      ,
00310           UTOL=1.0E-4 , VTOL=1.0E-4 , PTOL=1.E-4    , TTOL=1.0E-6  ,
00312         STRAT='NONE' , FLOW='INOUT', DELTAT=5.0    ,
00321      CONVEC='UPWIND', ENEREQ='YES', BOUNAC=1.0    ,
00320         PUNITS='ATM' , TUNITS='F'   , RITEXT='YES' , RITOUT='YES' ,
00330           STORIT=500   , STOINT=500   , MAP(1)='TYPE', RITINT='YES' ,
00330         RITERR='YES' , PRESAC=0.88 , FLOBAL='NO ' , MAPOUT='YES' $
00340 E$INPUT ITEM='BOUND' $
00340 E$INPUT ITEM='SLIP'   , I1=2   , J1=13   , I2=16   , J2=13 $
00340 E$INPUT ITEM='OUTLET' , I1=17 , J1=3 , I2=17 , J2=3  $
00370 E$INPUT ITEM='END' $
00380 E$INPUT ITEM='GENERAL' $
00390 E$INPUT ITEM='WIDTH'     , VALUE=3.0 , UNITS='FEET' $
00400 E$INPUT ITEM='VELX'      , VALUE=0.0  , UNITS='FPS' $
00400 E$INPUT ITEM='VELY'      , VALUE=0.0  , UNITS='FPS' $
00410 E$INPUT ITEM='TEMP'      , VALUE=70.6 , UNITS='F' $
00421 E$INPUT ITEM='COORD' , UNITS='FEET' $
00430 E$INPUT ITEM='END' $
00440 E$INPUT ITEM='SECTION' $
00465 E$INPUT ITEM='TEMP' , I1=1  , J1=2  , I2=1  , J2=6  ,
00465             VALUES = 6*62.0 , UNITS='F' $
00481 E$INPUT ITEM='WIDTH' , I1=1 , J1=1 , I2=1 , J2=13 ,
00481             VALUES = 13*1.0 , UNITS='FEET' $
00481 E$INPUT ITEM='WIDTH' , I1=2 , J1=1 , I2=2 , J2=13 ,
00481             VALUES = 13*1.5 , UNITS='FEET' $
00481 E$INPUT ITEM='WIDTH' , I1=3 , J1=1 , I2=3 , J2=13 ,
00481             VALUES = 13*2.0 , UNITS='FEET' $
00481 E$INPUT ITEM='WIDTH' , I1=4 , J1=1 , I2=4 , J2=13 ,
00481             VALUES = 13*2.5 , UNITS='FEET' $
00481 E$INPUT ITEM='WIDTH' , I1=5 , J1=1 , I2=5 , J2=13 ,
00481             VALUES = 13*3.0 , UNITS='FEET' $
00481 E$INPUT ITEM='VELX' , I1=1 , J1=2 , I2=1 , J2=7 ,
00481             VALUES = 6*0.0446 , UNITS='FPS' $
00481 E$INPUT ITEM='VELX' , I1=2 , J1=2 , I2=2 , J2=13 ,
00481             VALUES = 12*0.0149 , UNITS='FPS' $
00481 E$INPUT ITEM='VELX' , I1=3 , J1=2 , I2=3 , J2=13 ,
00481             VALUES = 12*0.0112 , UNITS='FPS' $
00481 E$INPUT ITEM='VELX' , I1=4 , J1=2 , I2=4 , J2=13 ,
00481             VALUES = 12*0.0089 , UNITS='FPS' $
00481 E$INPUT ITEM='VELX' , I1=5 , J1=2 , I2=5 , J2=13 ,
00481             VALUES = 12*0.0074 , UNITS='FPS' $
```

```
00481 E$INPUT ITEM='VELX' , I1=6 , J1=2 , I2=6 , J2=13 ,
00481         VALUES = 12*0.0064 , UNITS='FPS' $
00481 E$INPUT ITEM='VELX' , I1=7 , J1=2 , I2=7 , J2=13 ,
00481         VALUES = 12*0.0056 , UNITS='FPS' $
00481 E$INPUT ITEM='VELX' , I1=8 , J1=2 , I2=8 , J2=13 ,
00481         VALUES = 12*0.0050 , UNITS='FPS' $
00481 E$INPUT ITEM='VELX' , I1=9 , J1=2 , I2=9 , J2=13 ,
00481         VALUES = 12*0.0045 , UNITS='FPS' $
00481 E$INPUT ITEM='VELX' , I1=10 , J1=2 , I2=10 , J2=13 ,
00481         VALUES = 12*0.0041 , UNITS='FPS' $
00481 E$INPUT ITEM='VELX' , I1=11 , J1=2 , I2=11 , J2=13 ,
00481         VALUES = 12*0.0037 , UNITS='FPS' $
00481 E$INPUT ITEM='VELX' , I1=12 , J1=2 , I2=12 , J2=13 ,
00481         VALUES = 12*0.0034 , UNITS='FPS' $
00481 E$INPUT ITEM='VELX' , I1=13 , J1=2 , I2=13 , J2=13 ,
00481         VALUES = 12*0.0032 , UNITS='FPS' $
00481 E$INPUT ITEM='VELX' , I1=14 , J1=2 , I2=14 , J2=13 ,
00481         VALUES = 12*0.0030 , UNITS='FPS' $
00481 E$INPUT ITEM='VELX' , I1=15 , J1=2 , I2=15 , J2=13 ,
00481         VALUES = 12*0.0028 , UNITS='FPS' $
00481 E$INPUT ITEM='VELX' , I1=16 , J1=2 , I2=16 , J2=13 ,
00481         VALUES = 12*0.0026 , UNITS='FPS' $
00481 E$INPUT ITEM='VELX' , I1=17 , J1=3 , I2=17 , J2=3 ,
00481         VALUES = 0.0297 , UNITS='FPS' $
00510 E$INPUT ITEM='END' $
00520 /EOR
00530 /EOF
```

VECTOR

```
00100 /JOB
00110 MJLQXXX(T20,CM300,P02,STCA1)
00120 USER(XXXXXX,XXXXXX)
00130 ***   XVECTOR
00140 FETCH(DN=$BLD,GDN=OVECTOR,UN=XXXXXX,DS=CI)
00150 FETCH(DN=FT10,GDN=RSBCD0,DS=CI)
00160 FETCH(DN=FT11,GDN=RSBSOL6,DS=CI)
00170 ACCESS(DN=CCLIB,UN=EKSAPP)
00180 ASSIGN(DN=TAPE03,A=FT03)
00190 LDR(LIB=CCLIB,MAP=FULL,E=1)
00200 EXIT(U)
00210 COST(LO=F)
00220 EXIT(U)
00250 STORE(DN=TAPE03,GDN=TAPE03,UN=XXXXXX,DS=FF,DT=C)
00260 EXIT(U)
00270 LOGFILE(L=VPLTDAY)
00280 STORE(DN=VPLTDAY,UN=XXXXXX,DS=FF,DT=C)
00290 /EOR
00300 E$PARAM  NSTEPS=1 , S2=0.1 , SIZE=3.0 , SIZRAT=2.57 $
00320 /EOF
```

CONTUR

```
00100 /JOB
00110 MJLQXXX(T20,CM300,P02,STCA1)
00120 USER(XXXXXX,XXXXXX)
00130 ***   XCONTUR
00140 FETCH(DN=$BLD,GDN=OCONTUR,UN=XXXXXX,DS=CI)
00150 FETCH(DN=FT10,GDN=RSBCDO,DS=CI)
00160 FETCH(DN=FT11,GDN=RSBSOL6,DS=CI)
00170 ACCESS(DN=CCLIB,UN=EKSAPP)
00180 ASSIGN(DN=TAPE02,A=FT02)
00190 LDR(LIB=CCLIB,MAP=FULL,E=1)
00200 EXIT(U)
00210 COST(LO=F)
00220 EXIT(U)
00250 STORE(DN=TAPE02,GDN=TAPE02,UN=XXXXXX,DS=FF,DT=C)
00260 EXIT(U)
00270 LOGFILE(L=CPLTDAY)
00280 STORE(DN=CPLTDAY,UN=XXXXXX,DS=FF,DT=C)
00290 /EOR
00300 E$PARAM  NSTEPS=1 , SIZE=3.0 , SIZRAT=2.57 $
00360 E$INPUT  ITEM='TEMPERAT' , NUM=27 , UNITS='F' ,
00370  VALUES= 60.0 , 60.5 , 61.0 , 61.5 , 62.0 , 62.5 , 63.0 , 63.5 ,
00370          64.0 , 64.5 , 65.0 , 65.5 , 66.0 , 66.5 , 67.0 , 67.5 ,
00370          68.0 , 68.5 , 69.0 , 69.5 , 70.0 , 70.5 , 71.0 , 71.5 ,
00370          72.0 , 72.5 , 73.0 $
00400 E$INPUT  ITEM = 'END' $
00420 /EOF
```

APPENDIX E: NOTATION

B	lateral width
D	mass residual
e	internal (thermal) energy per unit mass
f	arbitrary function
g_i	components of gravity vector
J	$x_\xi y_\eta - x_\eta y_\xi$
n	superscript indicating time-step number
p	pressure
p_H	hydrostatic pressure
p_o	reference pressure
q_i	components of heat flux vector
T	temperature
t	time
u_i	components of fluid velocity
$\underset{\sim}{u}$	fluid velocity vector
u,v	x- and y-components of $\underset{\sim}{u}$, respectively
$\tilde{u},\tilde{v}$	fluxes of $\underset{\sim}{u}$ through surfaces of constant ξ and constant η , respectively
x,y	cartesian coordinates
α	$x_\eta^2 + y_\eta^2$
β	$x_\xi x_\eta + y_\xi y_\eta$
γ	$x_\xi^2 + y_\xi^2$
∇	gradient operator

∇^2	laplacian operator
Δ	incremental operator
μ	dynamic viscosity
κ	thermal conductivity
ρ	density
ρ_s	initial density
ρ_o	reference density
σ_{ij}	components of stress tensor
τ_{ij}	components of shear stress tensor
ξ,η	curvilinear coordinates
ω	acceleration parameter